

42
185 pts.
1985

AVON

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek





DISEÑADORES DE PANTALLAS



OMO ya hemos comentado anterior- mente, incluso dedicándole un capítulo entero al tema, la estética es uno de los puntos fuertes en el éxito de cualquier programa. A nadie se le oculta que esto se concreta principalmente en la pantalla, debido a lo cual, gran parte de los esfuerzos de programación se consumen en crear pantallas atractivas. Como quiera que esto no es en muchas ocasiones tarea fácil, los programadores han creado herramientas que les hagan algo más sencillo su trabajo. Los diseñadores de pantallas. Entre los diseñadores de pantallas existen lógicamente programas más o menos depurados, que se ajustarán a nuestro objetivo según lo complicado que sea éste. En general, un diseñador de pantallas es un programa que nos permite dibujar cómodamente por una pantalla limpia un punto que va dejando un rastro, a modo de lápiz levitadamente, y dado que la simple utilidad de esto es bien poca, se le van añadiendo otras características encaminadas a mejorar la calidad estética del resultado. Así, el lápiz es capaz de pintar en cualquiera de los colores del Spectrum, borrar, invertir, o pasar de un punto a otro sin afectar a lo ya escrito.

ZX DESIGNER (ABC SOFT)

Las características de estos programas se van complicando, aportando cada vez más herramientas para facilitar el trabajo de diseño. Por ejemplo, el programa ZX DESIGNER, además de las cualidades comunes mencionadas, incorpora las siguientes posibilidades:

- Fijar un punto que se utilizará como base en las opciones de trazado de figuras que a continuación veremos.
- Trazar un arco, para lo cual bastará con pulsar la tecla A e introducir la longitud del arco en radianes.
- Trazar un paralelogramo.
- Trazar una línea hasta la «punta» de nuestro lápiz.

- Trazar un círculo, estableciendo como centro el cursor.
- Copiar la pantalla en la impresora.
- Cambiar los atributos con los que se dibuja en la pantalla.
- Rellenar de color cualquier figura.
- Guardar la configuración de la pantalla actual en la memoria.
- Sombrar una figura, mediante el relleno con un trazo discontinuo.
- Variar el paso con que el «lápiz» se desliza por la pantalla, para así poder alterar su velocidad de desplazamiento.
- Cuadricular la pantalla, que será una función muy útil para determinar los atributos de cada posición de carácter.
- Borrar la pantalla.
- Grabar o cargar la pantalla en cinta.
- Hacer más grueso el trazo del «lápiz».
- Definir caracteres gráficos (UDG).
- Cambiar de banco de caracteres definidos (dispone de ocho bancos).
- Grabar o cargar UDGs.
- Escribir o cargar UDGs.

Aunque la pantalla de programación del ZX Designer no es especialmente espectacular, con este programa podremos conseguir buenos resultados.



- Escribir textos en la pantalla.
 - Borrar la última operación efectuada, función de extrema utilidad para los que somos demasiado ligeros de dedos.
 - Ampliar una zona de pantalla.
 - No alterar los atributos al paso del cursor.
- Como podemos comprobar es muy amplia la gama de efectos que se puede obtener con este programa. Pero existen aún ciertas exigencias que quedan sin cubrir, como por ejemplo la confección de figuras tridimensionales.

VU-3D (INVESTRONICA)

VU-3D es un sofisticado programa para dibujo tridimensional. Utilizando comandos sencillos podrá crear un objeto sólido, o conjunto de ellos, en el espacio tridimensional. Con este programa se puede obtener una imagen tridimensional del mundo haciendo rotar los objetos mirándolos desde diferentes ángulos.

Este mundo tridimensional se trata en conjunto con sus estructuras de datos, del mismo modo que una cámara programable, en el mismo sentido en que el ojo observa el mundo tridimensional. Podemos rotar un objeto y mirarlo desde lejos o de cerca desde diferentes direcciones. En este sentido, podemos diseñar un objeto sólido y obtener una sensación de su realidad tridimensional.



El **VRUN** permite crear sus construcciones más destacables con un panel de gráficos definibles (V.D.G.)



Justo en la realización de gráficos vectoriales puede ser contemplado por un observador de pantalla.



Para la confección de pantallas de este género contamos con la colaboración de los **DISEÑADORES DE PANTALLAS**

La función de creación permite, mediante un sencillo juego de comandos, construir un objeto en tres dimensiones. Una vez construido éste, existe la posibilidad de acercarlo o alejarlo en un efecto de zoom o girarlo en cualquier dirección y sentido; estas operaciones posteriores a la creación son efectuadas en una segunda fase denominada *display* (pantalla).

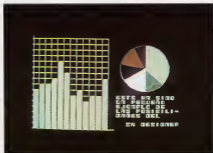
En una tercera etapa, habiéndose creado un objeto en la fase de creación y fijada su ampliación y punto de observación en la de *display*, podemos obtener, en un color de nuestra elección, el objeto en sus otras dos formas de presentación: *hidden line* (superficie frontal) o *shaded* (sombreado). Esta tercera etapa es la denominada *picture* (imagen).

Las opciones de la fase *picture* son: *shaded* (sombreado), *hidden line* (superficie frontal), *print* (imprimir), *colour* (color), *keep* (guardar), *quit* (salir). La función *colour* permite elegir el color apropiado en cada caso a la imagen representada, *print* realiza una copia en papel y *quit* nos devuelve a la fase *display*.

Keep es una opción muy a tener en cuenta, puesto que nos permite grabar la figura como una pan-

talla, la cual podrá ser cargada utilizando **SCREENS** en la presencia del programa **VU 3D**. El comando *hidden line* produce una imagen frontal de los objetos existentes en pantalla. Los extremos que limitan las caras superiores del objeto, que son aquellas que el observador puede ver, son los únicos líneas dibujadas, mientras que los de la cara posterior no se dibujan. La ejecución de este comando puede tardar un tiempo considerable si se trata de una figura complicada con muchas caras.

Otra forma de ver el objeto u objetos de forma tri-



El dibujo de pantalla no sólo se copia en el disco de los juegos, sino también en el *overmail*.



La cara B de la cinta de **ZX Designer** se encuentra con esta pantalla de presentación, seguida del programa en sí. La cara A está destinada a demostraciones.



dimensional es sombrear aquellos ceros que el observador puede ver. La claridad del sombreado de cada cara viene determinada por una combinación de luz de fondo y reflexión difusa desde una óptica fuente luminosa.

Así, utilizando el comando shade se nos interrogará sobre dónde deseamos situar la fuente de luz: arriba, centro o abajo, y además, izquierda, centro o derecha.

Además de las etapas mencionadas el programa nos permite modificar la figura diseñada o grabar, y por supuesto o cargar, las características

de la figura definida, no en forma de pantalla sino de fichero de datos. Finalmente, la opción abandonar (abandonar) nos permite borrar el fichero de datos existente, dejando la memoria libre para una carga desde casset, o crear una nueva figura en la primera etapa.

ARTIST (DINAMIC)

ARTIST es, sin duda, uno de los más completos equipos gráficos para el diseño de pantallas presentes en el mercado nacional. Su acción se concreta en tres puntos fundamentales: creación de figuras en tres dimensiones, definición de caracteres UDGs y diseño sobre la pantalla de alta resolución.



El programa H-3D nos permite obtener una imagen tridimensional haciendo rotar los objetos y mirando desde diferentes ángulos.



Al igual que la ampliación, H-3D permite también la reducción de la imagen dibujada.



Las diversas opciones del programa son combinables: en el ejemplo se ha empleado el sombreado y la ampliación.



En el denominado modo *plotter* se nos presenta una pantalla libre de 176 por 256 pixels en la cual podemos desplazar un cursor dejando un rastro a su paso, permitiendo dibujar dibujos sobre la pantalla. Este rastro vendrá dado en el color que nosotros indiquemos, pudiendo ser alterado éste en cualquier momento mediante la pulsación de las teclas numéricas del 0 al 7.

De modo similar podemos alterar el color de fondo y también de dos características tan importantes como el **FLASH** y el **BRIGHT**. Asimismo, también es posible definir el tamaño del cursor (pequeño o grande), o la velocidad de desplazamiento del mismo por la pantalla (tres velocidades). Naturalmente, también podemos desplazar el cursor borrando en vez de escribiendo, o simplemente dejando inalterado el contenido de la pantalla.

El programa también prevé la posibilidad de trazar líneas rectas, arcos circulares, o incluso círculos, representando sin duda una gran ayuda a la hora de confeccionar este tipo de figuras de cierta dificultad en pretender ser realizadas a mano alzada.

Asimismo, la posibilidad de hacer **AV** (rellenar

una superficie cerrada de un color determinado), acompañada del establecimiento de un **GRID**, conforman las características más destacables de este modo. El **grid** es una cuadrícula de colores que no afectan al dibujo pero nos ayudan, aprovechando la posibilidad de realizar *scrolls* de la pantalla, a realizar una correcta utilización del color en cada posición de carácter.

En este modo el programa **ARTIST** es muy similar a **MELBURN DRAW** que más adelante pasaremos a estudiar.

Sobre esta misma pantalla también es posible es-



Una de las opciones más interesantes del programa es la de *scrolling*, pudiendo elegir incluso la situación del foco de luz.



La opción de *Av* es la que nos permite la representación de las áreas rellenas del objeto.

enbr texto, incluso en caracteres gigantes, introduciendo el mensaje y el tamaño deseado para las letras.

En cuanto a la creación de gráficos definidos, disponemos de un diseño generador de UDGs muy similar a otros de este tipo, como el que acompañe la cinta de demostración de nuestro aparato. Así por ejemplo nos permite:

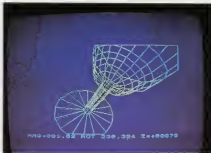
- Definir el gráfico en el carácter que queramos del banco seleccionado.
- Invertir el gráfico.
- Girar el gráfico a la derecha.

- Reflejar el gráfico.
- Borrar el gráfico.

El Artist también incorpora un tratamiento de imágenes tridimensionales, que podemos fundir una vez confeccionado con la pantalla principal. Para su diseño empleamos puntos definidos por tres coordenadas (X, Y y Z), que podemos dejar libres, o unir mediante una línea con el punto anteriormente definido. De forma similar al VU-3D, es posible acercar y alejar la figura, así como girarla en cualquier dirección y sentido.

MELBOURNE DRAW (MELBOURNE HOUSE)

Para finalizar vamos a hablar de un tipo de diseñador de tipo medio, que incorpora funciones si-



11. No conculpa la posibilidad de creación de las imágenes.



En el primer nivel de ZODIAC la pantalla se usa una multiplexión por centro.



El MELBOURNE DRAW incorpora funciones muy similares al modo plotter del ARTIST.



miles al modo *plotter* del Artist, y tiene como característica fundamental su sencillez de manejo: el MELBOURNE DRAW.

El programa se compone de un pequeño soporte BASIC, y un bloque de código máquina, que ejecuta propiamente el trabajo de diseño gráfico. El BASIC tan solo contiene un menú de opciones que da acceso al modo de edición y se ocupa de gestionar la grabación, carga y verificación de las pantallas o juego de gráficos definidos oreados.

Una copia de la pantalla se realiza sobre otra zona de memoria, de forma que podemos trabajar sin problema en las 24 líneas de la pantalla. Al entrar en el modo edición, un cursor representado por un punto de plot inmediatamente indica nuestra posición.

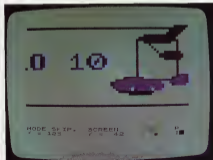
Para el desplazamiento del punto, utilizaremos las teclas que circundan la B (Q, W, E, A, D, Z, X y C), pudiendo encontrarse el cursor en cuatro modos: SKIP, SET, RESET o INVERT. El primero de ellos se selecciona mediante la pulsación de SPACE y nos permite desplazarnos sin afector a la pantalla, el segundo se activa mediante ENTER y hace que el punto vaya dejando rastro a su paso. El tercer modo (RESET) se utiliza para

borrar y se activa con la tecla O, mientras que la tecla I sitúa el modo INVERT que cambia de estado la pantalla en los puntos por los que pasa el cursor, es decir, apaga los puntos encendidos, y enciende los apagados.

Aunque el desplazamiento del cursor es nuestra herramienta básica para el dibujo, disponemos también de otras funciones de gran interés. En primer lugar, la pantalla va siempre acompañada de una zona de indicadores de dos líneas, que indican el lugar de la pantalla en que se encuentra el cursor (coordenadas X, Y), el modo de dibujo



La posibilidad de scroll punto a punto y un cualquier tamaño es otra de las características más interesantes del MELBOURNE DRAW.



En el segundo y último nivel de ZODD la pantalla aparece multipárrafo por abstracción.



(SKIP, SET, ETC.), los atributos empleados (fondo, tinta, brillo y parpadeo), y otras características que a continuación veremos. Esta zona, que habitualmente se sitúa en las dos últimas líneas de la pantalla (las reservadas al sistema), puede ser desplazada temporalmente a las dos primeras, pulsando CAPS SHIFT + 9, lo cual nos permite dibujar en las veinticuatro líneas de la pantalla.

El cambio de atributos se efectúa de una manera muy rápida, simplemente tenemos que pulsar la tecla numérica correspondiente al color deseado,

con CAPS SHIFT para la tinta y con CAPS SHIFT para el fondo, mientras que las teclas CAPS SHIFT + V afecta al modo FLASH y CAPS SHIFT + B a BRIGHT. Por otra parte, para hacernos una idea de conjunto, también podemos alterar el color del marco de pantalla, pulsando la tecla B y contestando a la pregunta BORDER? que aparecerá en la zona de indicadores con el código de color deseado.

Para efectuar acciones sobre la pantalla completa, deberemos pulsar CAPS SHIFT + R, lo cual nos dará acceso a un menú en la zona de indicadores para situar todo el fondo de pantalla del color fijado para fondo (opción F del menú), toda la tinta del color de la tinta (opción I), asignar fondo y tinta a la vez (BI), borrar la pantalla (S), borrar la pantalla y ajustar el color de fondo y tinta (AI) o bien regresar a la pantalla principal (N).

La asignación de color no sólo se efectúa en el momento de trazar el dibujo, sino que también puede ser establecida independientemente pulsando H. El cursor pasará a tener las dimensiones de un carácter, funcionando en los modos SET o SKIP, aunque en esta ocasión, en el modo SET no producirá modificación de la pantalla sino de los colores, tal como son establecidos mediante las teclas numéricas.

Para ayudarnos en la tarea de colorear, disponemos de un grú para orientarnos claramente sobre las posiciones de carácter en las que se encuentran los elementos del dibujo (tecla G) y de un sistema de zoom que nos puede ayudar a cuadrar las figuras en los caracteres que nos interesen, para ello pulsaremos la tecla K y desplazaremos la pantalla utilizando los mismos controles que para el cursor (teclas de alrededor de la S). Finalmente, el FILL (rellenado de superficies) que se consigue con la pulsación de CAPS SHIFT + F es otra característica de interés.

En lo referente al texto, también puede ser incluido libremente en la pantalla, pasándose a ese modo con la pulsación de T. En cuanto a los caracteres definidos, pueden ser también utilizados, disponiendo del modo GRAPHICS (CAPS SHIFT + 9) dentro del modo texto. La pulsación de U en el modo de pantalla principal, nos facilita la definición de cualquier UDG, con la cual la pantalla se convierte en una enorme pantalla de definición.

Por último, existe la posibilidad de ampliar y reducir temporalmente el contenido de la pantalla (M y N), para mayor detalle, o bien ampliar o reducir la pantalla definitivamente (CAPS SHIFT + B).

En el primer caso, se trata simplemente de un ZOOM de la zona de la pantalla en que nos encontramos, que puede ampliar la imagen cuatro u ocho veces. En el segundo caso se trata de una modificación permanente para ampliar el dibujo de toda la pantalla, con la consiguiente pérdida de parte de la información.



Una de las propiedades del MFL ROM VSE-DWH es aumentar, sin disminuir la pantalla, la situación de las posiciones de carácter para facilitar la asignación de color.



Además del aumento temporal (ZOOM) el MFL ROM VSE-DWH puede efectuar aumentos definitivos mediante la opción SCALL.



ANALIZANDO LOS GRUPOS



UCHOS quizá nos estemos preguntando cuáles son las extensas posibilidades del microprocesador de nuestro Spectrum. Somos bombardeados con una avalancha de nuevos conceptos y todavía no se ha justificado su utilidad.

Paciencia, fue necesario sentir unas sídidas bases antes de proseguir nuestro estudio del C/M. Hagamos un recorrido a través de los diferentes grupos de instrucciones para hacernos una idea bastante aproximada de las operaciones que puede ejecutar el Z 80.

Grupos de carga de 8 y 16 bits

RECORRIENDO LAS TABLAS

Volvamos a las tablas del capítulo anterior. Allí encontraremos el juego de instrucciones completo del Z 80 (casi 700 instrucciones diferentes), las cuales hemos subdividido en los siguientes bloques:

1) Carga de 8 bits: en este grupo se hayan encuadradas todas las instrucciones de intercambio de información entre registros individuales, y entre registros y posiciones de memoria.





!!

Las instrucciones de manipulación de bit constituyen el más numeroso grupo entre las nuevas implementadas en el microprocesador Z 80.

*

El grupo de entrada/salida permite la comunicación del microprocesador con los dispositivos externos a él.

*

No es posible dar una norma general que permita el acceso común a todos los grupos de instrucciones.

2) Carga de 16 bits: como observaréis, muy pocas transferencias de información pueden efectuarse entre pares de registros. Sin embargo, a este grupo pertenecen las instrucciones PUSH y POP, fundamentales en el manejo del STACK o pila.

3) Aritmético de 8 bits: las operaciones de suma y resta entre cantidades de 8 bits son ejecutadas por las instrucciones de este grupo. Es posible también, incrementar o decrementar en uno el contenido de un determinado registro o dirección de memoria.

4) Lógico: pertenecen a este bloque las instrucciones que hacen uso de los operadores AND, OR y XOR, discutidos en el capítulo dedicado al Álgebra de Boole, así como las comparaciones lógicas.

5) Aritmético de 16 bits: se trata de un grupo análogo al anterior pero efectuándose las operaciones entre pares de registros.

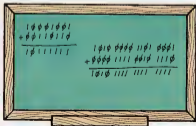
6) Aritmético y de control: comprende ciertas instrucciones de uso exclusivo del acumulador y el indicador de acarreo o acarreo, así como el manejo de las interrupciones.

Grupo lógico

7) Saltos y llamadas: permiten que el control del programa sea transferido a otra zona de ésta, incondicionalmente o en determinados supuestos establecidos previamente, volviéndose al punto donde se efectuó la bifurcación, en el caso de que la instrucción fuera de llamada.

Hemos incluido en este grupo las instrucciones de RESTART (RST), llamadas incondicionales a determinadas direcciones de la ROM, que tan solo ocupan un byte de memoria.

Grupos aritméticos de 8 y 16 bits



8) Retorno: las instrucciones de retorno permiten que el programa continúe su ejecución a partir del punto donde fue efectuada la última llamada. De la misma manera que en el grupo anterior, el regreso puede efectuarse según se cumplan o no algunas condiciones especiales.

9) Intercambio: en este grupo encontramos las únicas instrucciones del microprocesador Z80 que permiten el manejo del grupo de registros alternativos (ARS).

10) Transferencia: facilitan el movimiento de bloques de datos desde unas posiciones de memoria a otras.

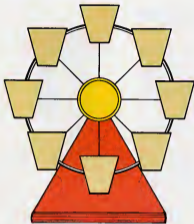
11) Búsqueda: permiten examinar un bloque de memoria con la intención de hallar un byte con el mismo contenido que el almacenado en el acumulador.

12) Tratamiento de bits: 240 nuevas instrucciones las cuales permiten manipular individualmente cada uno de los bits almacenados en cualquier registro o posición de memoria.

13) Rotación y desplazamiento: al igual que las anteriores, posibilitan la ejecución de operaciones a nivel de bit, dentro de cada octeto (byte).

14) Entrada/Salida: estas instrucciones habilitan al microprocesador para la comunicación con los dispositivos externos a él, recogiendo o enviando información desde o hacia ellos. Nuestra curiosidad debe estar ahora satisfecha. No dudaremos que con semejante volumen de instrucciones, las posibilidades de programación que ofrece nuestro microprocesador son francamente fabulosas.

Pero no hemos hecho sino comenzar, y todavía hemos de seguir ampliando conocimientos antes de pasar al análisis de cada instrucción en particular. El camino puede resultar largo y complicado a principio, pero pronto estaremos en condiciones de obtener los primeros resultados.



Grupo de rotación y desplazamiento

cada uno en particular, su utilidad se hará patente cuando estudiemos su forma de operar.

Por ejemplo, tomemos el grupo de carga de 8 bits. Entre paréntesis está señalado que en lenguaje ensamblador todas estas instrucciones comienzan por LD (abreviatura de Load, cargar).

Supongamos que buscamos la codificación de la instrucción LD R,A la cual significa que introducimos en el registro B el contenido del registro A.

En la parte superior de la tabla está señalado la palabra ORIGEN y a la izquierda DESTINO.

Por tanto, el destino es B y el remitente o fuente de la información es A.

Entrando en la tabla por la fila señalada con B, buscamos la intersección con la columna marcada con A y en la casilla correspondiente encontramos que el código de operación asociado al ensamblador LD R,A es 47h.

En otros grupos se define directamente el mínimo de la instrucción, como en el caso de los saltos y llamadas. Supongamos que la instrucción a codificar es CALL 7FFF, es decir una llamada a la subrutina ubicada a parte de la dirección 7FFF hexadecimal o 32767 decimal.

En la columna de la izquierda están los memem-

MANEJANDO LAS TABLAS

En las tablas podemos encontrar de manera rápida y clara los códigos de operación (en hexadecimal) correspondientes a todas las instrucciones implementables en el microprocesador de nuestro Spectrum.

Quizá nos encontremos algo sorprendidos por la anterior afirmación y no seamos capaces de ver en ellos otra cosa que un maremagnum de símbolos aparentemente sin sentido.

No es posible dar una norma general, la cual permita el acceso común a todas las guías, pues como descubriremos cuando sean analizados

!

Muestran las instrucciones de carga, son transferidas datos entre registro, o registros y posiciones de memoria.

*

El grupo lógico habilita los operandos AND, OR y XOR, así como también las comparaciones entre datos.

Grupo de transferencia

nicos correspondientes a estas instrucciones. En la sexta fila encontramos CALL. La llamada ha sido incondicional a la dirección especificada, a continuación seguirá el formato CALL n.n, siendo n los dos bytes que definen dicha dirección. Luego la codificación correcta será CD FF 7F. No nos sorprendamos si encontramos los dos bytes

Grupo de salto y llamada



con su posición intercambiada. Ya adelantamos en su momento, que el microprocesador cuando se trata de una dirección espera encontrar en primer lugar siempre el byte de menor peso. Otros símbolos utilizados son, por ejemplo, la «+» que representa el byte de desplazamiento en las instrucciones que manejan el direccionamiento indirecto, o la «-», la cual representa la extensión del salto en el direccionamiento relativo (los modos de direccionamiento serán el tema principal de nuestro próximo capítulo).

CARGADOR DE C/M

Para terminar el presente capítulo, hemos preparado un pequeño programa, el cual actuará de cargador hexadecimal para todas las rutinas en C/M con las que pondremos en práctica los ejemplos necesarios al explicar los distintos grupos de instrucciones.

La elección del sistema hexadecimal nos ha parecido la más adecuada, dada la extensa bibliografía existente que lo utiliza, en especial la literatura dedicada a la descripción de las rutinas almacenadas en la ROM.

Los códigos hexadecimales correspondientes a las instrucciones que configuran nuestro programa estarán incluidos en las sentencias DATA que situaremos al principio del programa.





y efectuando **RUN 8000**, quedarán almacenados en la memoria de nuestro ordenador para su posterior ejecución.

En caso de cometer algún error, se nos indicará la sentencia donde ocurrió para modificarla, mediante un mensaje intermitente que aparecerá sobre la pantalla del monitor.

Ejecute el modo mayúsculas al introducir los caracteres hexadecimales que conformen nuestra rutina, siendo siempre los dos últimos de esta **, de modo que el programa reconozca que no ha de seguir almacenando más código en la memoria.

Al comienzo se nos preguntará sobre la posición de memoria donde deseamos introducir la rutina en C/M. En principio, toda la RAM podría utilizarse, pero para evitar posibles conflictos con las Variables del Sistema y otras zonas dedicadas de ésta, elegimos posiciones de memoria cercanas al RAMTOP donde ejecutar nuestros programas, salvo que no se indique lo contrario.

Una vez almacenado el C/M es posible grabarlo utilizando el comando **BASIC SAVE "Nombre" CODE** dirección de inicio, longitud, siendo dirección de inicio la elegida por nosotros previamente y longitud, el número de bytes ocupados por la rutina (al finalizar la ejecución del programa cargador se nos informará también de este extremo en la pantalla).

A continuación encontraremos el listado del programa cargador. Tómate los cuidadosamente y grábelos antes de ponerlo en funcionamiento.

```

9999 READ AS,TOTAL
9100 FOR I=1 TO LEN AS-1 STEP 3
9110 IF ASC(I)>CHR$ 42 THEN LET M=I/100
    TO 9150
9120 LET H=ASC I TO E+1
9130 POKE DIR, FN B(H): LET DIR=DIR+1: L
    ET SUMA=SUMA+FN B(H): LET CONT=CONT+1
9140 NEXT I
9150 IF SUMA<>TOTAL THEN PRINT DE:DATA:
    CHR$ 32: FLASH 1:"CORROCA": GO TO 9970
9160 PRINT DE:DATA:CHR$ 32:"CORRECTA": C
    O TO 9970
    
```



Grupo de manipulación de bits



```

9980 REM #CARGADOR HEXADECIMAL#
9910 REM #CARLOS DE LA OSA VILLACIANO#
9920 DEF FN CHR$(CODE)=HEX$(CODE)
9930 DEF FN B(H)=16*FN D(H)+FN B(H)
    (2)
9940 LET DE="SENTENCIA DATA No. - "
9950 LET DATA=0: LET CONT=0: LET M=1
9960 INPUT "DIRECCION DE UBICACION ",DI
    R
9970 IF NOT M THEN PRINT : PRINT BRIGH
    T 3:"COMPLETADA LA CARGA": PRINT
    "LONGITUD, " (CONT) " BYTES": STOP
9980 LET SUMA=0: LET DATA=DATA+10
    
```



SALTO DE CABALLO

!

Según los directorios que requeridos al que sistema, el programa también más o menos función en su configuración, aunque para su mejorar su velocidad, hasta han sido eliminadas las llamadas a la subrutina de este módulo.

*

En caso de que no sea posible realizar un salto de caballo con las características que indiquemos, el programa nos lo hará saber mediante un mensaje.

*

Para la grabación del programa utilizaremos **SAVE "S CABALLO"**, o **SAVE "S CABALLO" LINE 1**, si deseamos escribir el sistema de autoejecución.



Es probable que muchos de nosotros compartamos una afición mutua: el sano deporte mental del «salto de caballo». Pues bien, con este programa podremos generar, a plena voluntad, todos los saltos de caballo que deseemos.

EL PROGRAMA

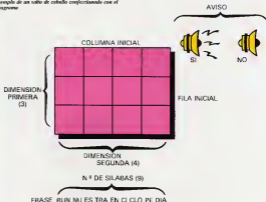
Este programa se compone de un soporte BASIC muy simple, que sirve para gestionar la toma de

datos y para la visualización del resultado generado por la auténtica inteligencia del programa: la rutina en código máquina.

Los datos que este programa para la confección del pasatiempo son los siguientes:

1. «FRASE»: a este INPUT debemos responder con la frase que deseamos utilizar en el salto de caballo, SEPARANDO SUS SILABAS MEDIANTE UN ESPACIO, y teniendo en cuenta que no son admisibles sílabas de más de tres letras.
2. «DIMENSION PRIMERA»: número de filas de que se compone el salto de caballo.
3. «DIMENSION SEGUNDA»: número de columnas de que se compone el pasatiempo.
4. «NUMERO DE SILABAS»: número de sílabas que se utilizan en el salto. Si este número es superior al total de sílabas que integran la frase, el cuadro del pasatiempo será completado con espacios vacíos.
5. «FILA INICIAL»: fila de la matriz elegida para

Ejemplo de un salto de caballo confeccionado con el programa.



el comienzo del salto. La primera columna es la
oera

6 «COLUMNA INICIAL» igual que «fila inicial»
pero referido a las columnas.

7 «AVISO S/N» avisado o no de un pitido inter-
mitente como aviso de que ha finalizado la con-
fección del salto de caballo.

La rutina en C/M actúa poniendo en práctica téc-
nicas recursivas, gracias a lo cual su ocupación
de espacio es mínima. A partir de la casilla de ori-
gen, el programa comprueba todas las series de

RUN ES ME JOR



*Los atletas que intente la frisa a tratar no deben
sublevarse en ningún caso los tres caracteres*



movimientos posibles, escogiéndolos de forma aleatoria.

Si al final de la secuencia elegida, no localiza un salto posible, da un paso atrás en su camino y comprueba otra nueva vía.

Puede el sistema de programación empleado, el tiempo de confección de un pasatiempo puede ser grande, si las características que hemos requerido de él así lo exigen. De igual modo, es posible que el programa llegue a la conclusión de la imposibilidad material de realizar el salto de caballo deseado, cuestión ésta que nos anunciará mediante la emisión del mensaje "NO existe el salto de caballo".

Para reducir el tiempo de confección del salto podemos emplear un pequeño truco: definir un cua-

drado con más casillas que soltabas tiene la tarea a tratar. De esta modo, se simplificará el labor de distribución por parte del programa, que completará las casillas sobrantes con tres asteriscos (***)

Para grabar el programa, basta con la utilización del siguiente comando **SAVE "S.CABALLO"**. Si preferiésemos la opción con autoejecución, teclearíamos la orden de grabación de la siguiente forma: **SAVE "S.CABALLO" LINE 1.**



Para confeccionar un salto de caballo es necesario exponer la fente a todas las casillas separadas por espacios

RUN NO ES TRA EN CI CLO PE DIA

```

1 DATA 221,489,199,129,321,489,721,126,489,281,221
,822,287,129,321,489,221,126,488,281
2 DATA 126,282,489,289,489,289,489,177,679
,486,488,321,432,387,126,221,489,281
3 DATA 227,481,197,129,489,167,129,481,485,488,327
,474,442,197,129,481,485,488,489,485
4 DATA 197,129,281,482,129,481,485,488,485,482
,227,486,424,197,129,481,167,129,481
5 DATA 485,488,227,176,281,462,255,458,191,129,489
,191,129,452,486,488,489,191,129,489
6 DATA 289,216,129,489,199,129,134,483,188,129,119
,285,229,129,489,189,129,134,483,187
7 DATA 129,119,254,488,274,192,138,483,193,129,158
,242,192,138,486,188,129,274,488,258
8 DATA 192,138,483,192,129,136,242,192,138,237,475
,187,129,285,255,129,221,126,488,214
9 DATA 488,191,192,138,483,187
10 DATA 192,138,483,191,129,136,242,192,138,237,475
,187,129,285,188,129,486,198,129,489,129
11 DATA 129,489,189,129,285,484,138,483,196,129,482
,325,422,138,489,195,129,254,680,194
12 DATA 192,138,237,475,167,129,285,225,129,221,486
,489,488,424,197,129,254,488,192,489
13 DATA 191,129,254,487,268,195,449,126
14 POKE 23656,0: FOR I=32229 TO 24864: READ XI: POKE
I,XI: NEXT I
15 DIM L$(22): DIM A$(2): DIM B$(256,2)
16 DATA 2,1,1,2,-1,2,-2,1,-2,-1,-2,1,-2,2,-1
17 RESTORE 281: FOR I=1 TO 81: READ A$(I): READ B$(I)
: NEXT I
18 GO TO 1888
19 GO SUB 2688
20 GO SUB 2525
21 PRINT AT 3,12: FLASH 1:"ESPERA"
22 RANDOMIZE USR 32229
23 IF D$(5) AND INC$(5) THEN GOTO .0,28: PAUSE
24 GO TO 45
25 GO SUB 1588
26 INPUT "QUIERES PROGRAM OTRA VEZ? (R) SI (N) NO"
: THEN GO TO 58
27 GO TO 18888
2888 REM INPUT FRASE
2898 PAPER 6: BORDER 6: CLS : INPUT "FRASE":(R) LET
FRASE=USR"488" : G
2908 LET CONT=0 : LET J=1
2918 LET I=J
2928 IF AR$(J)="" THEN LET J=J+1: LET I=J: GO TO 186
:
2938 IF AR$(J)="" THEN LET J=J+1: GO TO 1858
2948 LET C=AR$(I TO J)
2958 IF C=14 THEN RETURN
2968 LET B$(CONT)=C: LET CONT=CONT+1
2978 GO TO 1838
2988 REM SALTO
2998 GO SUB 2588

```

```

1518 IF NOT POKE 32219 THEN CLS : PRINT AT 10,2:"NO
EXISTE EL SALTO DE CABALLO" : RETURN
1528 LET Y1=INT (121-81*3/2)
1538 LET X1=INT (121-81*2/2)
1548 FOR I=0 TO N1-1
1558 FOR J=0 TO N1-1
1568 PRINT INT Y1:PAUSE 1:AR$(I+1)J)=INT (11+3J/2)
1578 NEXT J: NEXT I
1588 RETURN
1598 REM INICIALIZACION TABLAS
1608 FOR I=1 TO 81: INPUT FRASE(I)
1618 LET A$(I)=USR"488" : LET B$(I)=INT (FRASE(I)
1628 FOR L=1 TO 2: LET X$(L,I): LET A$(L)=USR"475" :
LET A$(L+1)=NEXT L
1638 NEXT I
1648 FOR I=0 TO 7
1658 FOR J=0 TO 7
1668 POKE 32229+I*8+J,11
1678 POKE 32231+I*8+J,12
1688 NEXT J
1698 FOR I=32247 TO 32253: POKE I,R: NEXT I
1708 RETURN
2588 REM PANTALLA
2598 BORDER 7: PAPER 5: CLS : PRINT PAPER 6:AT 6,R:IL
9:AT 21,R:IC
2608 FOR I=1 TO 21: PRINT PAPER 6:AT 1,R:1 "BAT 1,2"
: I: NEXT I: RETURN
2628 REM ENTRADA DATOS
2638 GO SUB 2588
2648 PRINT AT 6,3:"DIMENSION PRIMERA" : AT 6,3:"SEGUN
DA" : INPUT "NUMERO DE SILABAS" : AT 12,5
: FILE INITIAL : AT 14,5:"COLUMNA INICIAL" : AT 16,5:"
AVISO (5/M)"
2658 INPUT "DIMENSION PRIMERA" : (R)2: PRINT AT 6,3:R
2668 FOR I=0 TO N1-1: IF N1=0 OR N2=0 THEN GOTO .0,28:
GO TO 1818
2678 INPUT "DIMENSION SEGUNDA" : (R)1: PRINT AT 6,3:R
1: POKE 32217,M1: IF N1=0 OR N1=1 THEN GOTO .0,48:
GO TO 1828
2688 INPUT "NUMERO DE SILABAS" : (R)M: PRINT AT 10,2:
M: INPUT POKE 32216,M2: IF M2=0 OR M2=1 THEN GOTO .0,28:
GOTO 1818
2698 INPUT "FILAS DE LA CASILLA INICIAL" : (R)1: PRINT AT
12,2:R1: POKE 32214,Y1: IF Y1=0 OR Y1=1 THEN GOTO .0,
28: GO TO 1828
2708 INPUT "COL. DE LA CASILLA INICIAL" : (R)1: PRINT AT
14,2:R1: POKE 32213,X1: IF X1=0 OR X1=2 THEN GOTO .0,
28: GO TO 1828
2718 INPUT "AVISO (5/M)" : (R)5: PRINT AT 14,2:R5
2728 POKE 32219,M3: POKE 32226,M2: POKE 32221,R: POKE 3
2222,L4
2738 POKE 32877,FRASE(I)
2748 INPUT "ESTA BIEN ASÍ? (S/N)" : (R)S: IF R="N" THE
N GO TO 2528
2758 RETURN

```